

Code review

Eryk Tamm, nr indeksu 156087

Przedstawiony program realizuje podstawowe czynności związane z działaniem hotelu. Pozytywnie na czytelność kodu wpływa początkowa deklaracja klas:

```
class Hotel;  
class Room;  
class Guest;  
class Reservation;  
class Payment;
```

Wszystkie nazwy funkcji, zmiennych oraz stałych są jasno opisane, co umożliwia łatwe odczytanie kodu. Ponadto, kod realizujący sam program, napisany jest w całości w języku angielskim, co także wpływa na łatwiejsze zrozumienie kodu i jego działania. Zaimplementowane zostało także kilka przykładowych danych umożliwiających przetestowanie działania programu:

```
Hotel hotel("Sloneczny mlyn", "Portowa 5", "15:00:00", "  
10:00:00", "admin123");  
Room room1("101", "Standard", 200, 2);  
Room room2("102", "Deluxe", 300, 3);  
Room room3("103", "Standard", 300, 4);  
Room room4("104", "Deluxe", 600, 5);  
Guest guest1("Jan Kowalski", "jankowalski@gmail.com", "  
haslo123");  
Guest guest2("Pawel Nowak", "asd@gmail.com", "asd");
```

W tworzeniu programu zastosowano dobre praktyki programowania obiektowego, czyli dziedziczenie, na przykład klasa "CashPayment" dziedziczy po klasie "Payment". Spełniona jest także zasada polimorfizmu, metoda "processPayment()" używana jest zarówno przez klasę CardPayment jak i CashPayment:

```
class CardPayment : public Payment {  
public:  
    CardPayment(double amount) : Payment(amount) {}  
  
    void processPayment() override {  
        cout << "Platnosc karta. Kwota: " << amount << "  
z1\n";  
    }  
};  
class CashPayment : public Payment {  
public:  
    CashPayment(double amount) : Payment(amount) {}
```

```

    void processPayment() override {
        cout << "Płatność gotówka. Kwota: " << amount <<
            " zł\n";
    }
};

```

Program reprezentowany jest tylko tekstowo, dlatego nie jest on dostępny dla szerszej grupy osób. W niektórych miejscach brakuje komentarzy opisujących działanie kodu. Ponadto, w niektórych miejscach brakuje jasnych instrukcji dla użytkownika odnośnie wpisywanych danych wejściowych dla programu, przykładowo nie podano użytkownikowi formatu żądanej daty:

```

string inputDate() {
    string date;
    regex datePattern(R"(\d{4}-\d{2}-\d{2})"); // YYYY-MM-DD

    while (true) {
        cin >> date;
        if (regex_match(date, datePattern)) {
            return date;
        } else {
            cout << "Nieprawidłowy format daty. Spróbuj ponownie.\n";
        }
    }
}

```

Podsumowując, program spełnia najważniejsze zasady poprawnego programowania obiektowego. Potrzebne jest kilka usprawnień, które ułatwią odczytywanie samego kodu i korzystanie z programu.